

1

Getting started

7

Introduction	8
Addressing web pages	10
Defining document structure	12
Creating a HTML document	14
Validating documents	16
Employing a HTML editor	18
Summary	20

2

Providing page information

21

Bestowing a title	22
Specifying a character set	24
Limiting page life	26
Describing the document	28
Incorporating scripts	30
Incorporating style sheets	32
Linking more resources	34
Summary	36

3

Creating body content

37

Working the body	38
Inserting paragraphs	40
Including quotations	42
Proclaiming headings	44
Emphasizing text	46
Formatting text	48
Retaining preformatted text	50
Modifying text	52
Displaying code in text	54
Giving tooltip advice	56
Adding images	58
Summary	60

4

Adding style to content

61

Understanding style rules	62
Styling individual tags	64
Styling with style sheets	66
Styling with class	68
Styling by identity	70
Dividing content	72
Centering with style	74
Styling inline content	76
Summary	78

5

Writing lists

79

Creating unordered lists	80
Specifying bullet-point styles	82
Creating ordered lists	84
Specifying numbering styles	86
Creating definition lists	88
Styling definitions	90
Summary	92

6

Making tables

93

Producing a simple table	94
Spanning cells over rows	96
Spanning cells across columns	98
Adding a header & footer	100
Aligning cell content	102
Styling cell content	104
Asserting table frame rules	106
Grouping columns	108
Specifying column widths	110
Summary	112

7

Embedding objects

113

Embedding image objects	114
Embedding documents	116
Embedding Flash media	118
Embedding audio media	120
Embedding video media	122
Embedding QuickTime	124
Embedding RealPlayer	126
Summary	128

8

Inserting hyperlinks

129

Creating hyperlinks	130
Accessing links via keys	132
Linking to page fragments	134
Linking to protocols	136
Using images as hyperlinks	138
Producing image map links	140
Generating popups	142
Summary	144

9

Using frames

145

Framing in columns	146
Framing in rows	148
Creating a nested frameset	150
Hiding scroll bars	152
Changing frame content	154
Providing frame alternatives	156
Summary	158

10

Building forms

159

Submitting forms	160
Gathering text input	162
Providing text areas	164
Checking boxes	166
Choosing radio buttons	168
Selecting options	170
Uploading files	172
Utilizing hidden data	174
Pushing buttons	176
Using images for submission	178
Adding images to buttons	180
Labeling form controls	182
Scripting event attributes	184
Summary	186

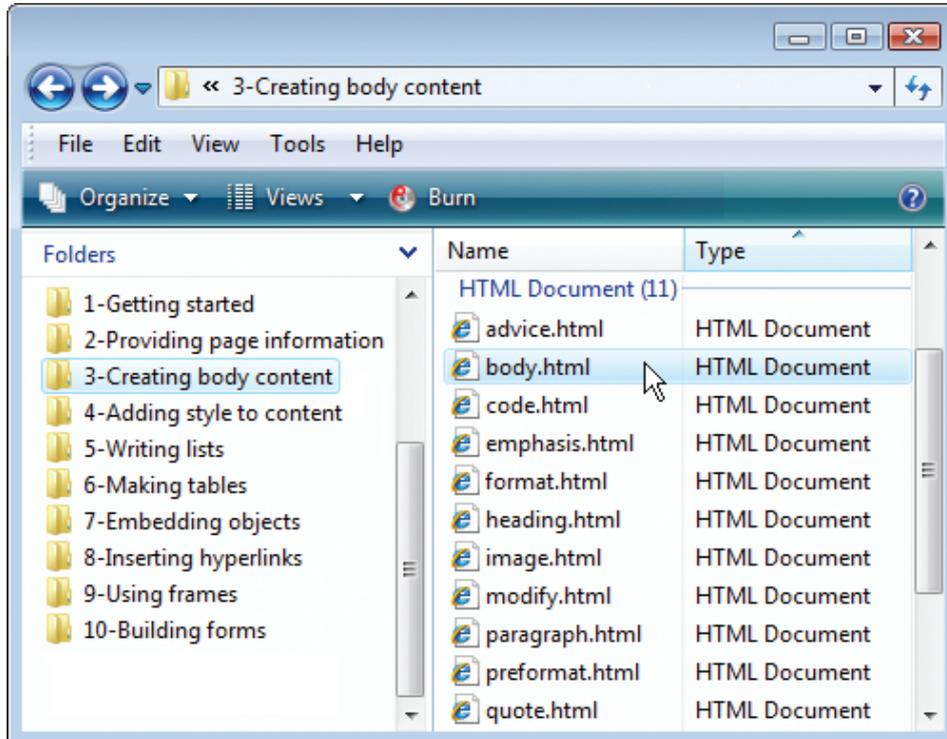
Index

187

Foreword

The examples in this book have been carefully prepared to demonstrate features of the HyperText Markup Language (HTML). You are encouraged to try out the examples on your own computer to discover the exciting possibilities offered by HTML. The straightforward descriptions should allow you to easily recreate the examples manually or, if you prefer, you can download an archive containing all the source code by following these simple steps:

- 1 Open your browser and visit our website at <http://www.ineasysteps.com>
- 2 Navigate to the “Resource center” and choose the “Downloads” section
- 3 Find the “From HTML in easy steps, 6th edition” item in the “Source code” list, then click on the hyperlink entitled “All code examples” to download the ZIP archive
- 4 Extract the contents of the ZIP archive to any convenient location on your computer – for easy reference these are arranged in sub-folders whose names match each chapter title of this book. The documents are named as described in the book and are located in the appropriate chapter folder of the archive. For example, the **body.html** document, described in the third chapter, is located in the folder named **3-Creating body content**



1

Getting started

*Welcome to the exciting
world of HTML.
This chapter introduces
HTML and demonstrates
how to create a “barebones”
HTML document.*

- 8** Introduction
- 10** Addressing web pages
- 12** Defining document structure
- 14** Creating a HTML document
- 16** Validating documents
- 18** Employing a HTML editor
- 20** Summary



The W3C is the recognized body that oversees standards on the web. See the latest developments on their informative website at www.w3.org.

Introduction

Historically, the desire to have text printed in specific formats meant that original manuscripts were “marked up” with annotation to indicate to the book-printer how the author would like sections of text laid out. This annotation had to be concise and needed to be understood both by the printer and the text originator. A series of commonly recognized abbreviations therefore formed the basis of a standard markup language.

HyperText Markup Language (HTML) is a modern standard markup language that uses common abbreviations called “tags” to indicate to the web browser how the author would like to have sections of a web page laid out. It was first devised in March 1989 by a British physicist named Tim Berners-Lee while he was working at CERN in Switzerland – the European Organization for Nuclear Research. His proposal showed how information could be easily transferred over the Internet using hypertext to provide the, now familiar, point-and-click navigation system. Connecting personal computers to the Internet using hypertext would create a single information network, allowing CERN physicists to share all computer-stored information at the laboratory. Berners-Lee created a text browser and in May 1990 the system was named the World Wide Web.

The simplicity of HTML led it to become popular in the early days of the Internet with text-based web browsers. A major development with HTML came in 1993 when a college student named Marc Andreessen added an image tag so that HTML could display images in addition to text. This version was then included in the Mosaic web browser from the National Center for Supercomputing Applications (NCSA) and became very successful. Marc went on to establish the Netscape web browser.

During the “browser wars” of the mid 1990s the various web browsers that were fighting for market share began to add proprietary tags to effectively create their own versions of HTML. The World Wide Web Consortium (W3C) standards organization recognized the danger that HTML could become fragmented and acted to create a standard to which all web browsers should adhere. Today Sir Tim Berners-Lee is a director of the W3C.

The latest recommended version of the W3C HTML standard is HTML 4.01. This version is covered throughout the book, but is generically referred to as “HTML 4” or just plain “HTML”.



Sir Tim Berners-Lee,
creator of
the World Wide Web

...cont'd

What's new in HTML 4?

The features in the earlier standard of HTML 3.2 are updated in HTML 4 to provide more flexibility for current and future development of the Internet. Most importantly HTML 4 separates document structure from document presentation. While HTML markup tags are still used to define document structure, controlling the layout of content, Cascading Style Sheets (CSS) are now used to define document presentation, controlling the appearance of content. Consequently many presentational tags have been made obsolete (deprecated) in favor of style sheets:

- The `<center>` tag that was used to center blocks is deprecated
- The `` and `<basefont>` tags that were used to specify fonts are deprecated
- The `<strike>`, `<s>`, and `<u>` tags that were used to decorate character appearance are deprecated
- The `<applet>` tag that was used to embed Java applets is now deprecated in favor of an `<object>` tag
- The `<dir>` and `<menu>` tags that were used for lists are deprecated in favor of a `` tag
- The `<isindex>` tag that was used to denote an input prompt is deprecated in favor of an `<input>` tag

HTML 4 introduces new tags to extend the precision of how document content may be defined and support has been added for languages written right-to-left to increase global utilization. Additionally a new model for table layout allows the browser to start rendering a table even before the entire table is downloaded.

HTML 4 has been designed to make web pages more accessible to those with physical limitations and greater scripting possibilities are provided by the addition of more page “events”. Further new tags allow the creation of compound web documents by the inclusion of resources from a variety of sources.

Don't forget



Deprecated features are still supported by “HTML 4 Transitional” and “HTML 4 Frameset” specifications for backward compatibility but not by the modern “HTML 4 Strict” specification described in this book.

Beware

A web page address (URL) cannot contain any blank spaces.

Addressing web pages

The World Wide Web comprises a series of large-capacity computers, known as “web servers”, which are connected to the Internet via telephone line and satellite. The web servers each use the HyperText Transfer Protocol (HTTP) as a common communication standard to allow any computer connected to any web server access to all files across the web.

HTML web pages are merely plain text files that have been saved with a file extension of “.htm” or “.html” – for instance **index.html**.

In order to access a file across the web its web address must be entered into the address field of a web browser. The web address is formally known as its “Uniform Resource Locator” (URL) and typically has three parts:

- **Protocol** – any URL using the HTTP protocol begins by specifying the protocol as **http://**
- **Domain** – the host name of the computer from which the file can be downloaded. For instance **www.example.com**
- **Path** – the virtual path to the file on the named domain, including any parent directory names where applicable. For instance **/htdocs/index.html**

A URL describing the location of a file by protocol, domain, and path is stating its “absolute” address. So the absolute address of the file described by the protocol, domain, and path components above is **http://www.example.com/htdocs/index.html**.

Code contained within a HTML web page can reference other HTML files in any domain by their absolute address. HTML files resident **within the same domain can also be referenced** by their absolute address or more simply by their “relative” address. This means that files located within the same directory can be referenced in another file just by their file name. For instance, a file named “adjacent.html” can be referenced as **adjacent.html**.

Additionally, a relative address can reference a file located within a parent directory by prefixing its file name with the syntax “../”. For instance, a file named “higher.html” in the parent directory can be referenced from the current directory as **../higher.html**.

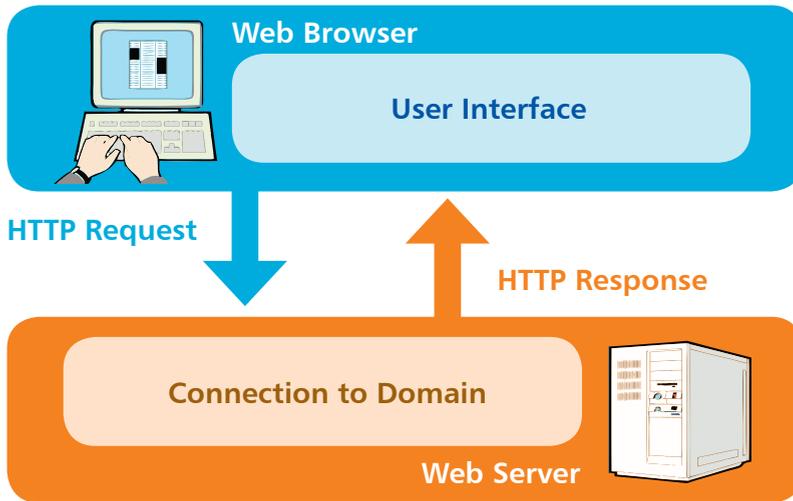
Hot tip

Where an address states only the HTTP protocol and a domain name most web servers are configured to seek a file named as **index.html** in their default directory.

...cont'd

How do web servers work?

When you enter a URL into the browser address field the browser first examines the specified protocol. Where the protocol is HTTP it recognizes that a file is being sought from a web server so then contacts a Domain Name Server (DNS) to look up the numerical Internet Protocol (IP) address of the specified domain. Next a connection is established with the web server at that IP address to request the file at the specified path. When the file is successfully located it is sent back to the web browser, otherwise the web server sends an error code such as “404 – Page Not Found”.



A successful response sends HTTP “headers” to the web browser, describing the nature of the response, along with the requested file. The headers are not normally visible but can be seen using special software, like the “ieHTTPHeaders” plugin for Internet Explorer:

```
ieHTTPHeaders x HTTP/1.1 200 OK
Content-type: text/html
Connection: Keep-Alive
Keep-Alive: timeout=150000, max=10
Transfer-Encoding: Chunked
Date: Mon, 18 Feb 2010 15:33:39 GMT
Server: Abyss/2.5.0.0-X1-Win32 AbyssLib/2.5.0.0
```

Here the web server identifies the protocol as HTTP/1.1 and sends the success code “200 - OK”.

Notice that the HTTP Content-type is described as “text/html”. This is the universally recognized unique MIME type that is used by all web servers to identify regular plain text HTML files.

Don't forget



The Domain Name Server is typically run by your Internet Service Provider or by your company.

Hot tip



You can learn more about the ieHTTPHeaders plugin at www.blunck.se or Linux users can invoke the `-i` option with `curl` to see the HTTP headers.

Beware

The Document Type Definition may appear on a single line if space permits but should otherwise be copied precisely as listed here.

Defining document structure

The structure of a HTML 4 document has these three parts:

- **Document Type Definition (DTD)** – defining precisely which HTML version type the document conforms to
- **Head section** – containing descriptive data about the document itself, such as the document’s title and character set
- **Body section** – containing the content that is to appear when the document gets loaded into a web browser

Document Type Definition

The Document Type Definition is a non-HTML tag that must appear at the start of the first line of every HTML 4 document. The DTD is just a piece of code, specified in the HTML standard, that can be pasted into the top of each page. It states an address containing the language rules for that version of HTML.

The standard DTD is called “Strict” and should be used for a HTML document that uses only valid HTML 4 markup. For instance, the DTD using HTML version 4.01 looks like this:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd" >
```

The specification also provides a “Transitional” DTD for backward compatibility. This should only be used where the HTML document contains some obsolete **deprecated markup**:

```
<!DOCTYPE HTML PUBLIC
    "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd" >
```

Finally there is a special DTD that is only to be used for a HTML “Frameset” document that specifies the frames to be used in a multi-document web page:

```
<!DOCTYPE HTML PUBLIC
    "-//W3C//DTD HTML 4.01 Frameset//EN"
    "http://www.w3.org/TR/html4/frameset.dtd" >
```

As the example code given in this book does not use any obsolete deprecated markup the Strict DTD is used in all example HTML documents except the special frameset examples in chapter nine.

Don't forget

The web browser can use the address in the DTD to validate the HTML code contained in that HTML document.

...cont'd

The entire head section and body section can be enclosed within a pair of `<html>` `</html>` tags to contain all HTML markup. The W3C HTML 4.01 specification actually states that these are optional but it is logical to provide a single “root” element. Most HTML tags are used in pairs like this to act as “containers” with the syntax `< tagname > data < /tagname >`

Head section

The document’s head section begins with a HTML opening `<head>` tag and ends with a corresponding closing `</head>` tag. Data describing the document can be added later between these two tags to complete the HTML document’s head section.

Body section

The document’s body section starts with an opening `<body>` tag and ends with a corresponding closing `</body>` tag. Actual document content that is to appear in the web browser can be added later between these two tags to complete the HTML document’s body section.

Code comments can be added at any point within both the head and body sections between a pair of `<!--` and `-->` tags. Anything that appears between these special comment tags is completely ignored by the web browser.

So the markup tags that create the fundamental document structure of a Strict HTML 4 document look like this:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd" >
<html>
  <head>
    <!-- Data describing the document to be added here. -->
  </head>
  <body>
    <!-- Actual document content to be added here. -->
  </body>
</html>
```

Note that HTML 4 tag names may appear in either uppercase or lowercase but future specifications may allow lowercase only – so only lowercase tag names are used throughout this book.

Hot tip



A HTML “element” is any matching pair of opening and closing tags, or any single tag not requiring a closing tag – as described in the HTML element tags list on the inside front cover of this book.

Hot tip



The “invisible” characters that represent tabs, newlines, carriage returns, and spaces are collectively known as “whitespace”. They may optionally be used to inset the tags for clarity.

Beware

HTML documents should not be created in Word processors, such as Word, as they include additional information in their file formats.



hello.html

Hot tip

The `<meta>` tag is a single tag – it does not need a matching closing tag. See the element tag list on the inside front cover of this book to find other single tags.

Creating a HTML document

The fundamental HTML document structure, described on the previous page, can be used to create a simple HTML document in any plain text editor such as Notepad on Windows systems. In order to create a valid “barebones” HTML document two tags need to be added to the head section – describing the document’s title, and the character set in which the document is written.

The document title is defined between `<title>` and `</title>` tags. Defining the character set requires values to be specified to two “attributes” within a HTML `<meta>` tag. The `http-equiv` attribute represents the equivalent of the HTTP Content-type header with the statement `http-equiv=“Content-type”`. The `content` attribute defines the MIME type as “text/html” and the encoding (`charset`) character set. Where the document is created in ANSI format for Western languages the encoding is “ISO-8859-1”.

Follow these steps to create a valid “barebones” HTML document :

1

Launch a plain text editor, such as Windows Notepad, and start a new document with the Document Type Definition describing the HTML 4 Strict version

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

2

After the DTD, create a root element

```
<html>
<!-- Head and body sections to be added here. -->
</html>
```

3

Within the root element insert a document head section

```
<head>
<!-- Descriptive information to be added here. -->
</head>
```

4

Within the head section, insert tags describing the document’s title

```
<title>Getting Started with HTML</title>
```

5

Within the head section, insert a tag describing the document’s character set

```
<meta http-equiv=“Content-type”
content=“text/html; charset=ISO-8859-1” >
```

...cont'd

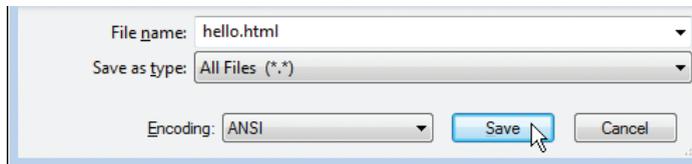
6 After the head section, insert a document body section
`<body>`
`<!-- Actual document content to be added here. -->`
`</body>`

7 Within the body section, insert a size one large heading
`<h1>Hello World!</h1>`

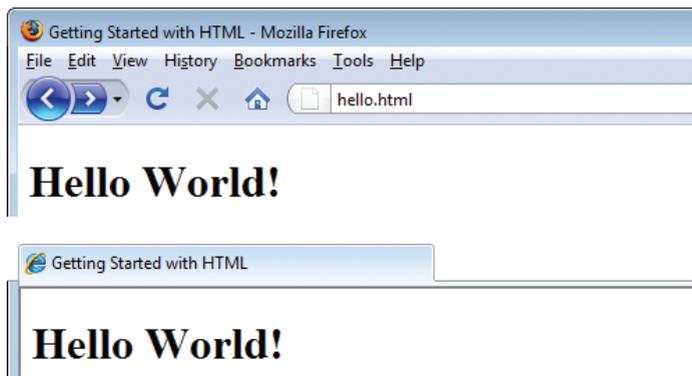


```
hello.html - Notepad
File Edit Format View Help
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd" >
<html>
<head>
<title>Getting Started with HTML</title>
<meta http-equiv="Content-Type"
content="text/html; charset=ISO-8859-1">
</head>
<body>
<h1>Hello world!</h1>
</body>
</html>
```

8 Save the document as “hello.html” being sure to set the encoding to ANSI



9 Now open the HTML document in a web browser to see the document title displayed on the title bar or tab and to see the document content displayed as a large heading



Don't forget 

Attribute values must always be enclosed within quotes.

Hot tip 

ANSI (American National Standards Institute) encoding is the “Latin 1” character set defined in the ISO-8859-1 standard.



The W3C's online HTML validator can be found at <http://validator.w3.org>.

Validating documents

Just as text documents may contain spelling and grammar errors HTML documents may contain various errors that prevent them from conforming to the specification rules. In order to verify that a HTML document does indeed conform to the rules of its specified Document Type Definition it can be tested by a validator tool. Only HTML documents that pass the validation test successfully are sure to be valid documents.

Web browsers make no attempt at validation so it is well worth verifying every HTML document with a validator tool before it is published, even when the content looks fine in your web browser. When the browser encounters HTML errors it will make a guess at what is intended – but different browsers can make different interpretations so may display the document incorrectly. Conversely, valid HTML documents should always appear correctly in any standards-compliant browser.

The World Wide Web Consortium (W3C) provide a free online validator tool that checks the syntax of web documents. It allows HTML documents that are found to be valid to include an icon demonstrating that you have taken the care to create a standards-compliant web page.

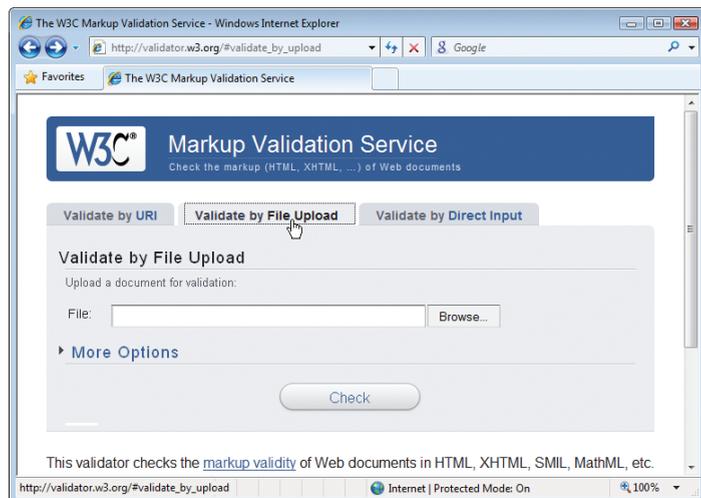
1

With an Internet connection, open your web browser and navigate to the W3C Validator Tool at validator.w3.org then click on the “Validate by File Upload” tab

Hot tip

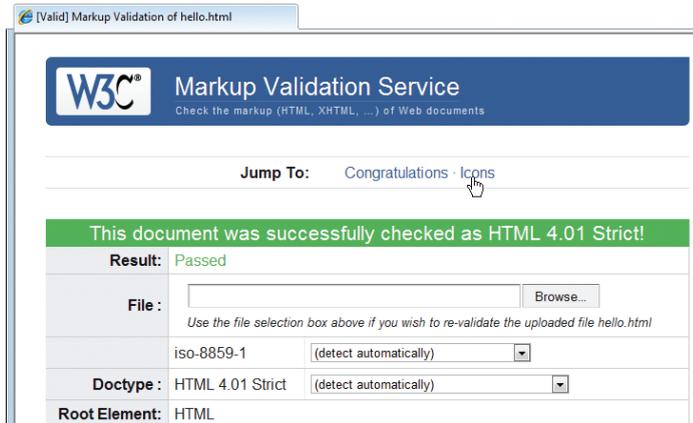


Other tabs in the validator allow you to enter the web address of a HTML document located on a web server to “Validate by URI” or copy’n’paste all code from a document to “Validate by Direct Input”.



...cont'd

- 2 Click the “Browse” button to open the Windows “Choose File” dialog then select your HTML document and click the Open button – the local path now appears in the validator’s File field
- 3 Click the validator’s “Check” button to upload a copy of the HTML document and run the validation test – the result will then be displayed



If validation fails the errors are listed so you may easily correct them. When validation succeeds you may choose to include a W3C validation icon at the end of the HTML document.

- 4 On the valid result page click the hyperlink to jump to the “Congratulations - Icons” section of the page
- 5 Copy’n’paste the HTML code of your preferred icon into the body section of your HTML document then save the document and open it in a web browser



Notice that the validator automatically detects the document’s character set and HTML version.



The W3C icon is also a hyperlink to the W3C validator tool – so the HTML document validation can be easily confirmed online.

Employing a HTML editor

HTML code can be created in any basic text editor and does not require any special software. In fact, many professional HTML authors use the simple Notepad application included with Microsoft Windows. As long as the code in the new text file is saved with a file extension of either “.htm” or “.html” a HTML document is created. This file can then be opened in any recent version of Microsoft Internet Explorer to see the browser output.

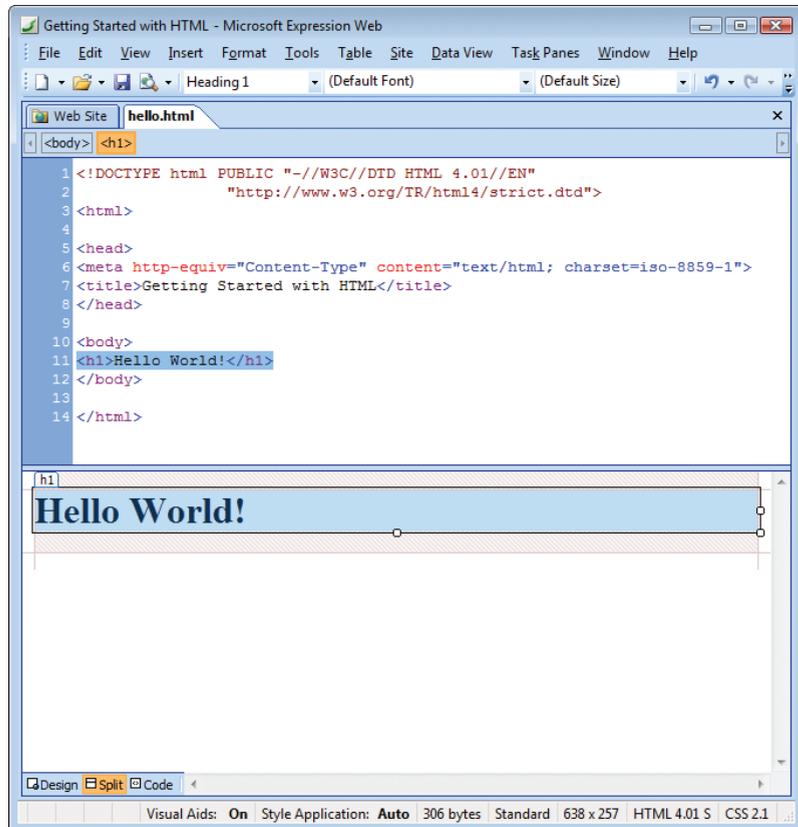
Some Netscape browsers and other older browsers may not fully recognize the HTML 4 specifications. It is best therefore to view HTML 4 web pages in a modern version of Internet Explorer or Mozilla Firefox.

Some HTML authors prefer to use specialized HTML editors that color-code the various parts of the HTML document for greater clarity. Microsoft’s Expression Web editor, shown below, provides many further features to help create great HTML code.

Beware

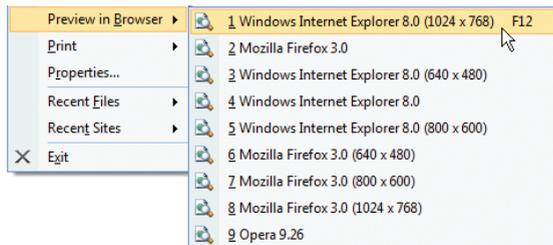


The Expression Web interface also allows web pages to be created visually by dragging components onto the Design window – but knowledge of HTML is often helpful in fine-tuning the web page.



...cont'd

The split view in Expression Web provides a live preview of what the HTML document will look like in a browser. Additionally a built-in menu lets you quickly view the document on any web browser installed on your computer in a variety of resolutions.



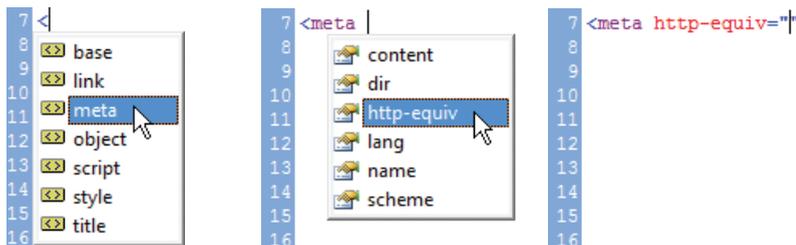
Don't forget



The W3C online validator tool can also be used to verify correct HTML code along with Expression Web's Compatibility Checker tool.

Expression Web's "Compatibility Checker" tool lets you easily locate code errors. For example, where a closing tag has been omitted the compatibility report provides the name of the unclosed tag and the line number where the error occurs.

Like other Microsoft development tools the Expression Web editor has "Intellisense" that provides context-sensitive menus, which can insert HTML tags compatible with the current point in the document. For example, when you type a "<" in the head section Intellisense presents a list of tags which may be inserted at that point. After selecting a tag and typing a space it then presents a list of attributes which may be inserted within that tag.



By default Intellisense also automatically inserts a matching closing tag, if appropriate, whenever you type an opening tag. Although intended to prevent code errors some authors find this feature can be annoying but the Expression Web options allow Intellisense features to be turned on and off individually. This means that the editor can be customized to your personal taste.

Hot tip



Expression Web provides Code Snippets to quickly insert frequently used chunks of code – Ctrl+Enter, dt4s inserts the entire HTML 4.01 Strict DTD.

Summary

- The World Wide Web Consortium (W3C) is the recognized body that oversees standards on the web
- In HTML 4, HTML element tags define document structure but Cascading Style Sheet (CSS) rules control its presentation
- HyperText Transfer Protocol (HTTP) is the common communication standard used by web servers
- Uniform Resource Locator (URL) is an absolute web address comprising Protocol, Domain, and Path components
- A relative address can reference an adjacent file by its file name and may use the syntax “../” to reference a parent directory
- Web servers send response headers back to the requesting computer along with the files requested, or an error code
- The MIME type describing HTML documents is “text/html”
- Each HTML 4 document must have a Document Type Definition, Head section, and Body section
- There are separate specifications for Strict, Transitional, and Frameset versions of HTML 4
- The `<html>` `</html>` tags create a root element container
- Information about the document itself is contained in the document head section, between `<head>` `</head>` tags
- Document content is contained in the document body section, between `<body>` `</body>` tags
- A HTML 4 document should state the document title between `<title>` `</title>` tags and its character set in a `<meta>` tag
- The free W3C online validator tool should be used to verify the document is indeed valid against the rules of its DTD
- HTML documents can be created in any plain text editor or in a specialized HTML editor like Microsoft Expression Web